**System Requirements Specification (SRS)** for the Job Application and Recruitment Web System:

**1. Project Overview**

This system is a job application and recruitment portal for a county public service board, built with:

* **Backend**: CodeIgniter 4 (CI4)
* **Frontend**: React + Vite with Ant Design components + Jsx and not tsx
* **DB** - Mysql

The system will support three distinct user roles:

* **Applicants**
* **Admins**
* **Board Committee**

It will feature public access pages and a role-based dashboard accessible upon authentication.

**2. Key Features by User Role**

**2.1 Public Users (Unauthenticated)**

Accessible without login:

* Home
* About
* Gallery
* FAQs
* Notices (from database)
* Job Listings (read-only)

Accessing any other resource triggers a **modal** from the right (slide-in drawer) prompting user to **Login / Signup**.

**2.2 Applicants**

**Account Features**

* Register via modal with OTP verification
* Login with email/password or phone + OTP
* Reset password

**Applicant Dashboard**

* Progress bar with checklist of steps:
  + Personal Details, Address, if employee
  + Educational Background
  + Short courses
  + Professionals
  + Employment History
  + Referees
  + Uploads (ID, academic, professional certificates, etc.)
* Each completed step updates the progress bar with ticks and a % completion

**Profile Management**

* Add/update personal details from applicants table
* Address: Select county > constituency > ward (dynamic dropdown)
* Include PWD info, KRA PIN, nationality, ethnicity, religion
* Save progress locally using DexieDB until submission

**Job Application**

* Browse available jobs
* System validates if applicant meets qualification
* Apply to a job (status: draft → submitted)
* View all applied jobs with status:
  + Submitted
  + Shortlisted
  + Interviewed
  + Rejected
  + Hired

**2.3 Admins**

**Admin Dashboard**

* Create/manage:
  + Counties
  + Constituencies
  + Wards
  + Courses
  + Specializations
  + Awards
  + Institutions
  + Departments
  + Professions
  + Designations
  + Jobs & their requirements
* View job applications and applicant profiles

**Job Management**

* Post job adverts with qualification filters (award, course, specialization), with a library that allows upload of either a word or pdf prepared advert and convert data to be submitted to the db
* Set application deadlines
* Enable/disable application submission

**Notifications**

* Send SMS notifications (shortlisted, interviewed, appointed) using external SMS API (Africa's Talking APIs)

**Reports**

* Generate reports (PDF, Excel)
  + Applications per job
  + Shortlisted candidates
  + Interview schedule
  + Final appointments

**2.4 Board Committee**

**Shortlisting Panel**

* View applicants by job and qualification match; be able to view attached pdf qualification documents in pdf iFrame;
* Mark applicant as:
  + Shortlisted
  + Interviewed
  + Appointed
  + Rejected

During interview, the shortlisting committee can see a list of shortlisted candidate their profile and award marks based on the parameters provided, see DB Table

* Add comments/remarks for each stage
* Assign interview date

**Interview Management**

* Set interview schedules
* Print interview sheets

**3. Functional Requirements**

| **Module** | **Description** |
| --- | --- |
| **Authentication** | Role-based login via modal, redirection to respective dashboards |
| **Applicant Registration** | Captures user info and initiates profile setup |
| **Profile Management** | Progressive form capturing all applicant details, saved step-by-step |
| **Dynamic Dropdowns** | Address hierarchy (County → Constituency → Ward) |
| **Course & Qualification Matching** | Based on course\_offered, specialization, award tables also in hierarchy |
| **Job Application** | Validation logic against job requirements before allowing apply |
| **Shortlisting** | Shortlist, schedule, comment, appoint through a committee interface |
| **Job Advert Posting** | Admin can create and publish jobs with qualification filters |
| **Reporting** | Auto-generate printable reports for each phase |
| **Notifications** | SMS integrations for status alerts |
| **Public Pages** | Static: About, FAQ, Gallery. Dynamic: Notices, Job Listings |

**4. Non-Functional Requirements**

| **Requirement** | **Specification** |
| --- | --- |
| **Performance** | Optimized React with Vite for fast load times |
| **Scalability** | Modular backend (CI4 services) and frontend components |
| **Security** | Role-based access, XSS/CSRF protection, input validation |
| **Usability** | Guided steps, progress indicators, mobile responsive |
| **Maintainability** | Clean MVC backend, component-based frontend |
| **Offline Support** | DexieDB used to store applicant progress locally |

**5. Database Entities & Relationships**

**Core Tables** from schema:

* users, applicants, applications, jobs, designations, departments, awards, courses, course\_offered, specializations, counties, constituencies, wards, professions, referees, institutions

Key relations:

* An applicant belongs to a user
* A job has requirements defined by award, course, specialization
* An application links a job and an applicant
* Dynamic lookups for dropdowns from relational tables

**6. UI/UX Notes**

* Ant Design Steps to show applicant progress
* Right-side modal for auth (Ant Design Drawer)
* Table interfaces (Ant Design Table) for jobs, applicants, reports
* Lazy-loaded components (React Suspense)
* Responsive design (flex/grid layout)

**7. Future Enhancements(Give a room for expansion of the system for the following)**

* Role-based access for interviewers vs. general committee
* Applicant notification via email
* PDF export for submitted applications
* Online interview scheduling integration
* Audit logs and activity tracking
* Records management portal
* Accounts management portal
* Wealth Declarations Portal
* Interns
* Attachees